ROOT Statistical Software

L. Moneta, I. Antcheva, R. Brun, A. Kreshuk
CERN, Geneva, Switzerland

Abstract
Advanced mathematical and statistical computational methods are required by the LHC experiments for analyzing their data. Some of these methods are provided by the ROOT project, a C++ Object Oriented framework for large scale data handling applications. We review the current mathematical and statistical classes present in ROOT, emphasizing the recent developments.

1 ROOT Math Work Package

The ROOT MATH work package is responsible to provide and to support a coherent set of mathematical and statical libraries required for simulation, reconstruction and analysis of high energy physics data. Existing libraries provided by ROOT are in the process of being re-organized in a new set of mathematical libraries with the aim to avoid duplication, increase modularity and to facilitate support in the long term. The main library components are the followings and shown in figure 1.

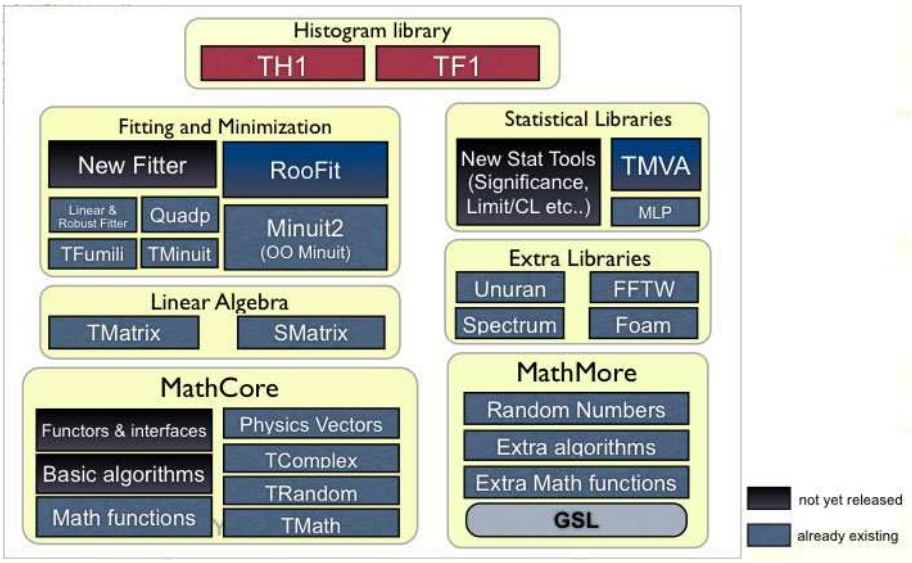

Fig. 1: New structure of the ROOT Mathematical Libraries. A different color code is used to distinguish components already existing from those which are in the process of being developed.

- **MathCore**: a self-consistent minimal set of mathematical functions and C++ classes for the basic needs of HEP numerical computing.
- **MathMore**: a package incorporating functionality which might be needed for an advanced user (as opposed to MathCore which addresses the primary needs of users) and dependent on external libraries like the GNU Scientific Library [1].
- **Linear Algebra**: vector and matrix classes and their related linear algebra functions. Two libraries exist: a general matrix package completed with a large variety of linear algebra algorithms and SMatrix, a dedicated package for small and fixed size matrices with optimal performances.
- **Fitting and minimization libraries**: classes and libraries implementing various types of fitting and function minimization methods, like Minuit and the new object-oriented version Minuit2.

– **Statistical libraries**: packages providing various algorithms for multi-variate analysis or classes for computing confidence levels and discovery significances using frequentist or Bayesian statistics.
– **Histogram libraries**: advanced classes for displaying and analyzing one, two and three dimensional data. It provides the histograms and profiles classes. Multi dimensional data sets are handled by the tree library.

In the following sections a detailed description is given for some of these components which have been recently developed and released. A brief description will be given also for those components that are planned to be introduced in ROOT.

2 Mathematical functions

New mathematical functions have been added recently in the MathCore and MathMore library to complement the functions existing in the namespace TMath and present in the ROOT core library. The new special functions are those proposed in the next extension of the C++ Standard Library [2] and follow the same naming scheme. These functions include all the major special functions, like the gamma, beta, error functions and also Bessel functions, hypergeometric functions, elliptic integrals, Legendre and Laguerre polynomials. Furthermore the MathCore and MathMore libraries provide all the major statistical distribution functions such as normal, χ^2, Cauchy, etc.., in a coherent naming scheme. For each statistical function, the probability density function, with suffix `_pdf`, (for example `normal_pdf` for the normal distribution), the cumulative distribution function with suffix `_cdf`, the complement of the cdf with suffix `_cdf_c`, the quantile function (inverse of the cdf), with suffix `_quantile`, and the inverse of the complement of the cdf, with suffix `_quantile_c` are provided.

Extensive tests of these functions have been performed [3] by comparing the numerical results obtained with the functions from other packages like Mathematica or Nag [4]. Often an accuracy at the level of 10^{-16} (double numerical accuracy) is reached for these functions.

3 Random Numbers

In ROOT pseudo-random numbers can be generated using the `TRandom` classes. A base class provides the methods for generating uniform and non-uniform numbers (according to specific distributions) while the derived classes, `TRandom1`, `TRandom2` and `TRandom3` implement pseudo-random number generators. These classes have been recently improved by replacing some obsolete generators. The following pseudo-random number generators are currently provided:

– Mersenne and Twister generator [5] implemented in the class `TRandom3`. This is the default generator in ROOT and the recommended one for the very good random propriety and its speed. It can also be seeded automatically using a 128 bit UUID number in order to generate independent streams of random numbers.
– RanLux generator [6] provided by the class `TRandom1`.
– Tausworthe generator [7] from L'Ecuyer provided by the class `TRandom2`. This generator has the advantage to use only 3 words of 32 bits for its state.

The CPU time results for generating a pseudo-random number using the ROOT generators are shown in table 3.

The base class `TRandom` provides also a Linear Congruential Generator. This generator has a state of only 32 bits and therefore a very short period and should not be used in any statistical application. `TRandom` implements as well methods for generating random numbers according to specific distributions. Recently a new faster algorithm for generating normal distributed random numbers, based on the acceptance-complement ratio method (ACR) [8], has been added to ROOT. This algorithm is much faster

Random Number Generator	Intel 32	Intel 64
MT (`TRandom3`)	22 ns	9 ns
Tausworthe (`TRandom2`)	17 ns	6 ns
RanLux (`TRandom1`)	120 ns	98 ns

Table 1: CPU time (in nanoseconds) for generating one pseudo-random number on a Linux box with the 32 or 64 bit architecture running CERN Scientific Linux 4 and using the GNU gcc version 3.4 compiler

than the traditional Box-Muller (polar) method used previously in ROOT which requires the evaluation of mathematical functions like `sqrt` or `log`. For example, on a 64 Intel Linux box running ROOT compiled with gcc 3.4, the time for generating one random gaussian number has been decreased from 183 to 42 ns.

The latest releases of ROOT contains in addition an interface to UNU.RAN [9], a software package for generating non-uniform pseudo-random numbers. It contains universal (also called automatic or black-box) algorithms that can generate random numbers from large classes of continuous (in one or multi-dimensions), discrete distributions, empirical distributions (like histograms) and also from practically all standard distributions. Efficient methods based on Markov-Chain Monte Carlo are as well provided for multi-dimensional distributions.

4 Numerical Algorithms

New numerical algorithms based on the GNU Scientific Library (GSL) [1] are provided by the MathMore library. Classes for numerical differentiation, various adaptive and non-adaptive integration, interpolation, minimization and root finding algorithms for one-dimensional functions are currently present. Algorithms for multi-dimensional functions like Monte Carlo integration and minimizations are in the process of being added. Fast Fourier Transforms are as well provided via an interface to the FFTW [10] package. The new algorithms are designed by presenting a single interface to the user for the various implementations. Alternative implementations which can be present in different libraries can then be loaded at run-time using the plug-in manager system.

5 Minimization and Fitting

Fitting in ROOT is possible directly via the `Fit(..)` methods of the various data object classes like histograms (classes `TH1`, `TH2`, `TH3`), graphs (classes `TGraph`, `TGraphErrors`, `TGraphAsymmErrors` and `TGraph2D`) and trees (class `TTree`). Methods like least-squares or binned and un-binned likelihood fits are supported. An interface class, `TVirtualFitter` exists to perform more sophisticated fits and to interface the minimization packages, like Minuit [11], Fumili [12] or Minuit2 [3]. In the case of linear fits, a dedicated class `TLinearFitter` exists to solve the resulting linear system. An extension to the linear fitter (robust fitter) for removing bad observations, outliers, based on the approximate Fast Least Trimmed Squares (LTS) regression algorithm for large data sets [13] exists as well. More complex fits can be performed by using the RooFit package [14], which is now distributed within ROOT.

A new object-oriented version of Minuit has been recently developed and it is now integrated inside ROOT as a new package, called Minuit2. It provides and enhances all the functionality of the original version. The profits from basing on an object oriented design are increased flexibility, easy maintainability in the long term and opening to extensions such as integration of new algorithms, new functionality and changes in user interfaces. For example, the Fumili algorithm has been integrated directly inside the minimization framework provided by Minuit2. Various extensive tests have been performed to study and validate the numerical quality, convergence power and computational performances of this new version. In the future it is expected to improve the functionality by adding the possibility of supplying constraints on the parameters.

A new GUI for fitting has been introduced in order to drive the fitting process. It is possible to select the fitting function, to set the initial parameter values, fitting and minimization options with possibility of choosing the minimization engine. It is foreseen to be improved soon by adding advance drawing functionality such as contour plots, residuals and confidence levels.

In the future it is planned to improve the existing ROOT fitting classes, by extending the functionality of the `TVirtualFitter` class, by providing support for parallel fits, various fitting and minimization methods and easier integration with RooFit.

6 Statistical tools

For multi-variate analysis and signal-background discrimination a new package, TMVA [15], has been integrated recently in ROOT. It provides various algorithms, like automatic cuts optimizations, likelihood estimators, neural networks and boosted decision trees with common interfaces to use them easily together. Neural networks can also be used directly via the class `TMultiLayerPerceptron`. `TMultiDimFit` is another multi-dimensional method present in ROOT, which provides the possibility to find the parametrization of multi-dimensional functions with polynomials, Chebyshev or Legendre functions. It is used for example to parametrize the LHCb magnetic field from the measured field map. The class `TPrincipal` gives the possibility to perform principal component analysis to reduce dimensionality of the data while keeping as much information as possible. The class `TRobustEstimator` implements the Minimum Covariance Determinant estimator, a robust technique [16] to find the location and scatter of multi-dimensional data.

For estimating confidence levels, the class `TFeldmanCousin` computes upper limits for Poisson processes in the presence of background using the Feldman-Cousin method [17]. The class `TRolke` computes again the confidence intervals for Poisson processes but including the treatment of uncertainties in the background and in the signal efficiency using a profile likelihood method [18]. The class `TLimit` computes instead the confidence intervals using the CL_s method used for LEP Higgs searches [19]. It is applied to histograms representing the data, the simulated signal and the background and it incorporates the systematic uncertainty using a Bayesian approach.

A new package is also currently being developed to extend and improve the functionality of estimating confidence levels to satisfy the LHC requirements and focusing in particular on estimating discovery significances. It will both include frequentists and Bayesian methods and it will be based on the RooFit data modeling framework [20]. Tools for easy statistical combinations of results will be as well provided by this new package.

7 Conclusions

ROOT contains already a large variety of mathematical and statistical functionality required for the analysis of LHC data. An effort is on-going to consolidate and improve the existing libraries by replacing obsolete algorithms, by making them easier to use and by improving their modularity to gain in long term maintainability. The needs and the feedback received from users working on data analysis and reconstruction of the experiment data are as well taken into account in this consolidation process. Many of the statistical tools currently present in ROOT have been developed by various contributors from the high energy physics community. It is therefore important to ensure a continuation of these user contributions and to provide as well an easy way for the users to plug-in their developed tools. This consolidation effort should as well aim to remove duplications and provide implementations which are considered standard by the community.

References

[1] M. Galassi et al, *The GNU Scientific Library Reference Manual* - Second Edition, ISBN = 0954161734 (paperback). See also the url `http://www.gnu.org/software/gsl`

[2] W. Brown and M. Paterno, *A proposal to Add Mathematical Special Functions to the C++ Standard Library*, WG21/N1422 = J16/03-0004, available at the url `http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2004/n1687.pdf`.

[3] M. Hatlo *et al.*, *IEEE Transactions on Nuclear Science* **52-6**, 2818 (2005)

[4] The Numerical Algorithm Group (Nag) C Library, see the url `http://www.nag.co.uk/numeric/cl/CLdescription.asp`

[5] M. Matsumoto and T. Nishimura, *Mersenne twister: A 623-dimensionally equidistributed uniform pseudorandom number generators*, ACM Trans. on Modeling and Computer Simulations, 8, 1, (1998), 3-20

[6] F. James, *RANLUX: A Fortran implementation of the high quality pseudo-random number generator of Lüscher*, Computer Physics Communication, 79 (1994) 111.

[7] P. L'Ecuyer, *Maximally Equidistributed Combined Tausworthe Generators*, Mathematics of Computation, 65, 213 (1996), 203-213

[8] W. Hoermann and G. Derflinger, *The ACR Method for generating normal random variables*, OR Spektrum 12 (1990), 181-185.

[9] see the url `http://statistik.wu-wien.ac.at/unuran`.

[10] see the url http://www.fftw.org.

[11] F. James, *MINUIT Reference Manual*, CERN Program Library Writeup D506.

[12] S. Yashchenko, *New method for minimizing regular functions with constraints on parameter region*, Proceedings of CHEP'97 (1997).

[13] P.J. Rousseeuw and K.Van Driessen, *Computing LTS Regression for Large Datasets*, *Estadistica* **54**, 163 (2002).

[14] see the url `http://roofit.sourceforge.net`.

[15] F. Tegenfeld *et al.*, *TMVA - Toolkit for multivariate data analysis with ROOT*, proceedings to this workshop, see also the url `http://tmva.sourceforge.net`.

[16] P.J. Rousseeuw and K.Van Driessen, *A fast algorithm for the minimum covariance determinant estimator*, *Technometrics* **41**, 212 (1999).

[17] G.J. Feldman and R.D. Cousins, *Unified approach to the classical statistical analysis of small signals*, *Phys.Rev.* **D57**, 3873 (1998).

[18] W. Rolke, A. Lopez, J. Conrad, *Nuclear Instruments and Methods* **A551**, 493-503 (2005).

[19] T. Junk, *Nuclear Instruments and Methods* **A434**, 435-443 (1999).

[20] W. Verkerke, *Statistical software tools for LHC analysis*, proceedings of this workshop, 2007.